

# Counting Inversions Cheatsheet

## Problem:

Problem: Given an array of  $N \leq 10^5$  elements  $a_1, \dots, a_N$ , count the number of inversions. An inversion is a pair of elements  $i$  and  $j$  such that  $i < j$  and  $a[i] > a[j]$ .

Example:

$N = 7$

34, 6, 23, 0, 5, 99, 2

Answer:

There are 13 inversions.

## Implementation:

C++:

```
int ft[N+1];
void update(int x, int v) {while(x<=N) ft[x]+=v, x+=(x&-x);}
int query (int x) { return x>0 ? ft[x]+query(x-(x&-x)):0;}
```

Java:

```
int[] ft = new int[N+1];
public static void update(int x, int v) {while(x<=N) {ft[x]+=v; x+=(x&-x);} }
public static int query (int x) { return x>0 ? ft[x]+query(x-(x&-x)):0;}
```

Both languages:

```
// update (int x, int v): array[x] += v
// query (int x): return arr[1] + ... + arr[x]

// beforehand coordinate compress the values
inversions = 0; // this should be a long
for (int i=0; i<N; i++) {
    // update answer
    inversions += query (N) - query (value[i]);
    // update stored information with the current value
    update (value[i], 1);
}
```

## Runtime:

$O(N * \log N)$